EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp	
S69	496	(collect\$3 near garbage) and (mark\$3 near object\$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR ON		2007/08/29 17:18	
S70	2	(collect\$3 near garbage) and (mark\$3 near object\$1) and (unmark\$3 near card\$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:19	
S71	45	(collect\$3 near garbage) and concurrent and (mark\$3 near object\$1) and (unmark\$3 near object\$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	USPAT; USOCR; EPO; JPO; DERWENT;		2007/08/29 17:23	
S72	21	((unmark\$3 with object) same (allocation or reallocation))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:26	
S73	592	(707/206).CCLS.	USPAT; USOCR	OR	OFF	2007/08/29 17:25	
S74	54	S73 and (collect\$3 near garbage) and concurrent and (mark\$3 near object\$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:38	
S75		S72 and S73	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:26	
S76	27	S73 and (collect\$3 near garbage) and concurrent and (mark\$3 near object\$1) and unmark\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:30	

EAST Search History

			Ţ	· · · · · · · · · · · · · · · · · · ·	,	·
S77	0	S73 and (unmark\$3 with card) and ((unmark\$3 with object) same (allocation or reallocation))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:30
S78	16	(collect\$3 near garbage) and concurrent and (mark\$3 near object\$1) and (unmark\$3 near object\$1) and card\$1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR ·	ON	2007/08/29 17:30
S79	41	((card\$1 or page\$1) near (mark\$3 or dirt\$3)) with ((object\$1) near (mark\$3 or dirt\$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:30
S80	3	((card\$1 or page\$1) near (mark\$3 or dirt\$3)) and ((card\$1 or page\$1) near (unmark\$3 or un\$1mark\$3 or not\$1dirt\$3)) and (((card\$1 or page\$1) near (unmark\$3 or un\$1mark\$3 or not\$1dirt\$3)) with object\$1) and (((card\$1 or page\$1) near (mark\$3 or dirt\$3)) with object\$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:31
S81	2	(trac\$3 or find\$3) with ((card\$1 or page\$1) near (unmark\$3 or un\$1mark\$3 or not\$1dirt\$3)) with object	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:31
S82	20	(collect\$3 near garbage) and concurrent\$2 and (object\$1 with (pointer\$1 or link\$1)) and (new\$3 with object\$1) and ((unmark\$3 with object\$1) same allocat\$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:32
S83	27	S73 and (collect\$3 near garbage) and concurrent and (mark\$3 near object\$1) and unmark\$3	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/29 17:38

may help in reducing the length of the delay time, they cannot reduce the ...

	Go	Ó	O	Ó	O	O	O	Ö	Ü	US	ξĺ	e	
Result Page:										<u>10</u>	_		

concurrent garbage collection

Search Patents

Google Patent Search Help | Advanced Patent Search

Google Home - About Google - About Google Patent Search
©2007 Google

Sign in



concurrent garbage collection

Search Patents

Advanced Patent Sear Google Patent Search

Patents

Patents 1 - 10 on concurrent garbage collection. (0.17 seconds)

Real time, concurrent garbage collection system and method

US Pat. 5088036 - Filed Jan 17, 1989 - Digital Equipment Corporation ... concurrent garbage collection ...

Computer memory system with parallel garbage collection independent from an associated user ...

US Pat. 4775932 - Filed Jul 31, 1984 - Texas Instruments Incorporated

The term "concurrent" processing is used to mean garbage collection processing which

... Concurrent garbage collection, for instance, includes time shared ...

Method and system for eliminating synchronization between sweep and allocate in a concurrent ...

US Pat. 6289360 - Filed Oct 7, 1998 - International Business Machines Corporation

[3] Damien Doligez, Xavier Leroy, A concurrent generational garbage collector

... In mark-sweep garbage collectors, garbage collection is implemented in two ...

Database management system for controlling concurrent access to a database

US Pat. 4627019 - Filed Jul 8, 1982 - AT&T Bell Laboratories

10 The operation of the **garbage collection** method may 15 index entry in head block 501-6 is being considered, all be better understood by considering the ...

System and hardware module for incremental real time garbage collection and memory management

US Pat 5560003 - Filed Dec 21, 1992 - Iowa State University Research Foundation, Inc.

The garbage collection system's principal responsibilities are, ... During garbage collection, requests to read or write 65 memory that has not yet been ...

Adaptive memory management system for collection of garbage in a digital computer

US Pat. 5136706 - Filed Apr 30, 1987 - Texas Instruments Incorporated

Garbage is reclaimed in small amounts over a period of time, ideally without unduly degrading the overall system performance. Concurrent garbage collection ...

Method and system for concurrent garbage collection

US Pat. 6502111 - Filed Jul 31, 2000 - Microsoft Corporation

... concurrent garbage collection ...

Mostly concurrent compaction in a garbage collection system

US Pat: 6249793 - Filed Jun 10, 1999 - Sun Microsystems, Inc.

... concurrent ... garbage collection ...

Garbage collection without fine-grain synchronization

US Pat. 6052699 - Filed Dec 10, 1997 - Lucent Technologies Inc.

More particularly, **concurrent garbage collection** techniques fall generally into one of two classes: (1) variations on mark & sweep collectors, see, ...

Method and apparatus for assisting garbage collection process within a java virtual machine

US Pat 6070173 - Filed Nov 26, 1997 - International Business Machines Corporation

While existing software implementations of concurrent garbage collection processes

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: • The ACM Digital Library • The Guide

concurrent garbage collection

बनगात्रा

THE ACM DICITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used: concurrent garbage collection

Found 14,281 of 209,709

Sort results

by

results

relevance Display

expanded form

Save results to a Binder ? Search Tips Copen results in a new

Try an Advanced Search Try this search in The ACM Guide

Results 1 - 20 of 200

window

Result page: 1 2 3 4 5 6 7 8 9 10

Relevance scale \square

Best 200 shown

Correctness-preserving derivation of concurrent garbage collection algorithms

Martin T. Vechev, Eran Yahav, David F. Bacon

June 2006 ACM SIGPLAN Notices, Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06, Volume 41

Issue 6

Publisher: ACM Press

Full text available: Additional Information: full citation, abstract, references, index terms

Constructing correct concurrent garbage collection algorithms is notoriously hard. Numerous such algorithms have been proposed, implemented, and deployed - and yet the relationship among them in terms of speed and precision is poorly understood, and the validation of one algorithm does not carry over to others. As programs with low latency requirements written in garbagecollected languages become part of society's missioncritical infrastructure, it is imperative that we raise the level of confid ...

Keywords: concurrent algorithms, concurrent garbage collection, synthesis, verification

2 Concurrent replicating garbage collection

James O'Toole, Scott Nettles

July 1994 ACM SIGPLAN Lisp Pointers, Proceedings of the 1994 ACM conference on LISP and functional programming LFP '94, Volume VII Issue 3

Publisher: ACM Press

Full text available: pdf(919.87 KB)

Additional Information: full citation, abstract, references, citings, index terms

We have implemented a concurrent copying garbage collector that uses replicating garbage collection. In our design, the client can continuously access the heap during garbage collection. No low-level synchronization between the client and the garbage collector is required on individual object operations. The garbage collector replicates live heap objects and periodically synchronizes with the client to obtain the client's current root set and mutation log. An experimental implementation usi ...

Mostly concurrent garbage collection revisited

Katherine Barabash, Yoav Ossia, Erez Petrank

October 2003 ACM SIGPLAN Notices, Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications OOPSLA '03, Volume 38 Issue 11

Publisher: ACM Press

Full text available: pdf(279.42 KB)

Additional Information: full citation, abstract, references, citings, index terms

The mostly concurrent garbage collection was presented in the seminal paper of Boehm et

al. With the deployment of Java as a portable, secure and concurrent programming language, the mostly concurrent garbage collector turned out to be an excellent solution for Java's garbage collection task. The use of this collector is reported for several modern production Java Virtual Machines and it has been investigated further in academia. In this paper, we present a modification of the mostly concu ...

Keywords: JVM, Java, concurrent garbage collection, garbage collection, incremental garbage collection

4 Concurrent compacting garbage collection of a persistent heap

James O'Toole, Scott Nettles, David Gifford

December 1993 ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles SOSP

'93, Volume 27 Issue 5

Publisher: ACM Press

Full text available: pdf(1.50 MB)

Additional Information: full citation, abstract, references, citings, index terms

We describe a replicating garbage collector for a persistent heap. The garbage collector cooperates with a transaction manager to provide safe and efficient transactional storage management. Clients read and write the heap in primary memory and can commit or abort their write operations. When write operations are committed they are preserved in stable storage and survive system failures. Clients can freely access the heap during garbage collection because the collector concurrently builds a comp ...

5 Concurrent garbage collection using hardware-assisted profiling

Timothy H. Heil, James E. Smith

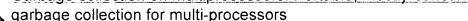
October 2000 ACM SIGPLAN Notices, Proceedings of the 2nd international symposium on Memory management ISMM '00, Volume 36 Issue 1

Publisher: ACM Press

Full text available: Pdf(1.74 MB) Additional Information: full citation, abstract, citings, index terms

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

6 Garbage collection on multiprocessors: Portable, mostly-concurrent, mostly-copying



Antony L. Hosking

June 2006 Proceedings of the 2006 international symposium on Memory management ISMM '06

Publisher: ACM Press

Full text available: 📆 pdf(196.04 KB) Additional Information: full citation, abstract, references, index terms

Modern commodity platforms increasingly support thread-level parallelism, which must be exploited by garbage collected applications. We describe the design and implementation of a portable *mostly-concurrent* mostly-copying garbage collector that exhibits scalable performance on multi-processors. We characterize its performance for heap-intensive workloads on two different multi-processor platforms, showing maximum pause times two orders of magnitude shorter than for fully stop-the-world co ...

Keywords: ambiguous-roots, concurrent, conservative, garbage collection, incremental, memory management, mostly-concurrent, mostly-copying, portability

7 Concurrency: Message analysis-guided allocation and low-pause incremental garbage collection in a concurrent language





Konstantinos Sagonas, Jesper Wilhelmsson

October 2004 Proceedings of the 4th international symposium on Memory management ISMM '04

Publisher: ACM Press

Full text available: pdf(650.12 KB)

Additional Information: full citation, abstract, references, citings, index terms

We present a memory management scheme for a concurrent programming language where communication occurs using message-passing with copying semantics. The runtime system is built around process-local heaps, which frees the memory manager from redundant synchronization in a multithreaded implementation and allows the memory reclamation of process-local heaps to be a private business and to often take place without garbage collection. The allocator is guided by a static analysis which speculative ...

Keywords: Erlang, concurrent languages, incremental and real-time garbage collection, thread-local heaps

8 Very concurrent mark-&-sweep garbage collection without fine-grain synchronization Lorenz Huelsbergen, Phil Winterbottom



October 1998 ACM SIGPLAN Notices, Proceedings of the 1st international symposium on Memory management ISMM '98, Volume 34 Issue 3

Publisher: ACM Press

Full text available: pdf(1.36 MB)

Additional Information: full citation, abstract, references, citings, index terms

We describe a new incremental algorithm for the concurrent reclamation of a program's allocated, yet unreachable, data. Our algorithm is a variant of mark-&-sweep collection that---unlike prior designs---runs mutator, marker, and sweeper threads concurrently without explicit fine-grain synchronization on shared-memory multiprocessors. A global, but infrequent, synchronization coordinates the per-object coloring marks used by the three threads; fine-grain synchronization is achieve ...

Concurrent garbage collection using program slices on multithreaded processors Manoj Plakal, Charles N. Fischer



October 2000 ACM SIGPLAN Notices, Proceedings of the 2nd international symposium on Memory management ISMM '00, Volume 36 Issue 1

Publisher: ACM Press

Full text available: 📆 pdf(957,62 KB) Additional Information: full citation, abstract, citings, index terms

We investigate reference counting in the context of a multi-threaded architecture by exploiting two observations: (1) reference-counting can be performed by a transformed program slice of the mutator that isolates heap references, and (2) hardware trends indicate that microprocessors in the near future will be able to execute multiple concurrent threads on a single chip. We generate a reference-counting collector as a transformed program slice of an application and then execute this slice in ...

10 A parallel, incremental, mostly concurrent garbage collector for servers

Katherine Barabash, Ori Ben-Yitzhak, Irit Goft, Elliot K. Kolodner, Victor Leikehman, Yoav Ossia, Avi Owshanko, Erez Petrank

November 2005 ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 27 Issue 6

Publisher: ACM Press

Additional Information: full citation, abstract, references, citings, index Full text available: pdf(683.50 KB) terms

Multithreaded applications with multigigabyte heaps running on modern servers provide new challenges for garbage collection (GC). The challenges for "server-oriented" GC include: ensuring short pause times on a multigigabyte heap while minimizing throughput penalty, good scaling on multiprocessor hardware, and keeping the number of expensive multicycle fence instructions required by weak ordering to a minimum. We designed and

implemented a collector facing these demands building on th ...

Keywords: Garbage collection, JVM, concurrent garbage collection

11 New garbage collection algorithms and strategies: Garbage-first garbage collection



David Detlefs, Christine Flood, Steve Heller, Tony Printezis

October 2004 Proceedings of the 4th international symposium on Memory management ISMM '04

Publisher: ACM Press

Full text available: pdf(199.59 KB)

Additional Information: full citation, abstract, references, citings, index terms

<i>Garbage-First</i> is a server-style garbage collector, targeted for multi-processors with large memories, that meets a soft real-time goal with high probability, while achieving high throughput. Whole-heap operations, such as global marking, are performed concurrently with mutation, to prevent interruptions proportional to heap or live-data size. Concurrent marking both provides collection "completeness" and identifies regions ripe for reclamation via compacting evacuation. This ev ...

Keywords: concurrent garbrage collection, garbage collection, garbage-first garbage collection, parallel garbage collection, soft real-time garbage collection

12 Garbage collection on multiprocessors: Task-aware garbage collection in a multi-





tasking virtual machine

Sunil Soman, Laurent Daynès, Chandra Krintz

June 2006 Proceedings of the 2006 international symposium on Memory management ISMM '06

Publisher: ACM Press

Full text available: 📆 pdf(125.22 KB) Additional Information: full citation, abstract, references, index terms

A multi-tasking virtual machine (MVM) executes multiple programs in isolation, within a single operating system process. The goal of a MVM is to improve startup time, overall system throughput, and performance, by effective reuse and sharing of system resources across programs (tasks). However, multitasking also mandates a memory management system capable of offering a quarantee of isolation with respect to garbage collection costs, accounting of memory usage, and timely reclamation of heap reso ...

Keywords: java, multi-tasking, resource reclamation, task-aware garbage collection, virtual machine

13 A concurrent copying garbage collector for languages that distinguish (im)mutable



data

Lorenz Huelsbergen, James R. Larus

July 1993 ACM SIGPLAN Notices, Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '93,

Volume 28 Issue 7

Publisher: ACM Press

Full text available: pdf(921.14 KB)

Additional Information: full citation, abstract, references, citings, index

This paper describes the design and implementation of a concurrent compacting garbage collector for languages that distinguish mutable data from immutable data (e.g., ML) as well for languages that manipulate only immutable data (e.g., pure functional languages such as Haskell). The collector runs on shared-memory parallel computers and requires minimal mutator/collector synchronization. No special hardware or operating system support is required.

14 A generational mostly-concurrent garbage collector Tony Printezis, David Detlefs





October 2000 ACM SIGPLAN Notices, Proceedings of the 2nd international symposium on Memory management ISMM '00, Volume 36 Issue 1

Publisher: ACM Press

Full text available: pdf(1.67 MB)

Additional Information: full citation, abstract, citings, index terms

This paper reports our experiences with a mostly-concurrent incremental garbage collector, implemented in the context of a high performance virtual machine for the Java™ programming language. The garbage collector is based on the "mostly parallel" collection algorithm of Boehm et al., and can be used as the old generation of a generational memory system. It overloads efficient write-barrier code already generated to support generational garbage collection to also ident ...

15 Concurrency: Write barrier elision for concurrent garbage collectors



Martin T. Vechev, David F. Bacon

October 2004 Proceedings of the 4th international symposium on Memory management ISMM '04

Publisher: ACM Press

Full text available: pdf(490.73 KB)

Additional Information: full citation, abstract, references, citings, index terms

Concurrent garbage collectors require write barriers to preserve consistency, but these barriers impose significant direct and indirect costs. While there has been a lot of work on optimizing write barriers, we present the first study of their elision in a concurrent collector. We show conditions under which write barriers are redundant, and describe how these conditions can be applied to both incremental update or snapshot-at-the-beginning barriers. We then evaluate the potential for write b ...

Keywords: concurrent garbage collection, write barrier

16 Garbage collection of actors



Dennis Kafura, Doug Washabaugh, Jeff Nelson

September 1990 ACM SIGPLAN Notices, Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications OOPSLA/ECOOP '90, Volume 25 Issue 10

Publisher: ACM Press

Full text available: pdf(791.19 KB)

Additional Information: full citation, abstract, references, citings, index

This paper considers the garbage collection of concurrent objects for which it is necessary to know not only "reachability," the usual criterion for reclaiming data, but also the "state" (active or blocked) of the object. For the actor model, a more comprehensive definition than previously available is given for reclaimable actors. Two garbage collection algorithms, implementing a set of "coloring" rules, are presented and their computational comp ...

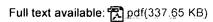
17 Garbage collecting the Internet: a survey of distributed garbage collection



Saleh E. Abdullahi, Graem A. Ringwood

September 1998 ACM Computing Surveys (CSUR), Volume 30 Issue 3

Publisher: ACM Press



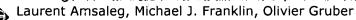
Full text available: 🛱 pdf(337.65 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Internet programming languages such as Java present new challenges to garbagecollection design. The spectrum of garbage-collection schema for linked structures distributed over a network are reviewed here. Distributed garbage collectors are classified first because they evolved from single-address-space collectors. This taxonomy is used as a framework to explore distribution issues: locality of action, communication overhead and indeterministic communication latency.

Keywords: automatic storage reclamation, distributed, distributed file systems,

distributed memories, distributed object-oriented management, memory management, network communication, object-oriented databases, reference counting

18 Garbage collection for a client-server persistent object store



August 1999 ACM Transactions on Computer Systems (TOCS), Volume 17 Issue 3

Publisher: ACM Press

Full text available: pdf(267.18 KB)

Additional Information: full citation, abstract, references, citings, index terms, review

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environment. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does non require callbacks to clients. It is fault-tolerant, but performs very little logging. The algorithm has been designed to be integrated into existing systems, and therefore it works with standard i ...

Keywords: client-server system, logging, persistent object-store, recovery

19 A concurrent, generational garbage collector for a multithreaded implementation of



🏟 MI

Damien Doligez, Xavier Leroy

March 1993 Proceedings of the 20th ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '93

Publisher: ACM Press

Full text available: pdf(1.01 MB)

Additional Information: full citation, abstract, references, citings, index terms

This paper presents the design and implementation of a "quasi real-time" garbage collector for Concurrent Caml Light, an implementation of ML with threads. This two-generation system combines a fast, asynchronous copying collector on the young generation with a non-disruptive concurrent marking collector on the old generation. This design crucially relies on the ML compile-time distinction between mutable and immutable objects.

20 Real-time replication garbage collection



Scott Nettles, James O'Toole

June 1993 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation PLDI '93, Volume 28 Issue 6

Publisher: ACM Press

Full text available: pdf(1.05 MB)

Additional Information: full <u>citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, index terms

We have implemented the first copying garbage collector that permits continuous unimpeded mutator access to the original objects during copying. The garbage collector incrementally replicates all accessible objects and uses a mutation log to bring the replicas up-to-date with changes made by the mutator. An experimental implementation demonstrates that the costs of using our algorithm are small and that bounded pause times of 50 milliseconds can be readily achieved.

Keywords: concurrent collection, copying garbage collection, incremental collection, realtime garbage collection, replication

Results 1 - 20 of 200

Result page: 1 2

2 3 4

56

8 9

next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat OuickTime Windows Media Player Real Player

Home | Login | Logout | Access Information | Alerts | Purchase History |

Welcome United States Patent and Trademark Office

□ Guest Search Results

BROWSE

SEARCH

IEEE XPLORE GUIDE

Results for "(concurrent garbage collection) <in> metadata"

Your search matched 8 of 1637503 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

.... e-πail

Login

Username

Password

» Forgot your password?

Please remember to log out when you have finished your session.

» Key

Indicates full text access

IEEE JNL

IEEE Journal or

Magazine

. IET JNL

IET Journal or Magazine

IEEE CNF

IEEE Conference

Proceeding

IET CNF

IET Conference

Proceeding

IEEE STD IEEE Standard

Article Information

1. Successive approximation of abstract transition relations

Das, S.; Dill, D.L.;

Logic in Computer Science, 2001. Proceedings, 16th Annual IEEE Symposium on

16-19 June 2001 Page(s):51 - 58

Digital Object Identifier 10.1109/LICS.2001.932482

Abstract | Full Text: PDF(564 KB) | IEEE CNF

Rights and Permissions

2. Relational profiling: enabling thread-level parallelism in virtual machines

Heil, T.; Smith, J.E.;

Microarchitecture, 2000. MICRO-33. Proceedings. 33rd Annual IEEE/ACM Internation:

10-13 Dec. 2000 Page(s):281 - 290

Digital Object Identifier 10.1109/MICRO.2000.898078

Abstract | Full Text: PDF(844 KB) | IEEE CNF

Rights and Permissions

Hardware support for concurrent garbage collection in SMP systems

Chang, J.M.; Srisa-An, W.; Chia-Tien Dan Lo;

High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fou

Conference/Exhibition on

Volume 1, 14-17 May 2000 Page(s):513 - 517 vol.1

Digital Object Identifier 10.1109/HPC.2000.846607

Abstract | Full Text: PDF(396 KB) IEEE CNF

Rights and Permissions

Distributed/concurrent garbage collection in distributed shared memory system:

Kordale, R.; Ahamad, M.; Shilling, J.;

Object Orientation in Operating Systems, 1993., Proceedings of the Third International

9-10 Dec. 1993 Page(s):51 - 60

Digital Object Identifier 10.1109/IWOOOS.1993.324927

Abstract | Full Text: PDF(792 KB) | IEEE CNF

Rights and Permissions

Implementing orthogonal persistence: a simple optimization using replicating co

Nettles, S.; O'Toole, J.;

Object Orientation in Operating Systems, 1993., Proceedings of the Third Internationa

9-10 Dec. 1993 Page(s):177 - 181

Digital Object Identifier 10.1109/IWOOOS.1993.324909

Abstract | Full Text: PDF(336 KB) IEEE CNF

Rights and Permissions

Improving the Performance of the Deferrable Server Based Garbage Collection 5

El Desokey, Ali Ebrahim; El Gawad, Aida Abd; Sarhan, Amany; Moawed, Seham; Information & Communications Technology, 2006, ICICT '06. ITI 4th International Con-Dec. 2006 Page(s):1 - 2

Digital Object Identifier 10.1109/ITICT.2006.358273

Abstract | Full Text: PDF(98 KB) | IEEE CNF

Rights and Permissions

7. Hardware concurrent garbage collection for short-lived objects in an object-orie Yu Wing Shing; Li, R.; Fong, A.S.;

Electrical, Electronic and Computer Engineering, 2004. ICEEC '04. 2004 International 5-7 Sept. 2004 Page(s):285 - 288

Abstract | Full Text: PDF(752 KB) IEEE CNF Rights and Permissions

8. Concurrent and distributed garbage collection of active objects Kafura, D.; Mukherji, M.; Washabaugh, D.M.;

Parallel and Distributed Systems, IEEE Transactions on Volume 6, Issue 4, April 1995 Page(s):337 - 350 Digital Object Identifier 10.1109/71.372788

Abstract | Full Text: PDF(1308 KB) · IEEE JNL Rights and Permissions

Help Contact Us Privacy & .

© Copyright 2006 IEEE –

Indexed by ज्ञि Inspec